REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
	Final	1 Jul 07 - 30 Nov 2011

4. TITLE AND SUBTITLE
Final Report: Autonomic Recovery of Enterprise Wide Systems After Attack of Failure With Forward Corrections

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-07-1-0527

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jiang Wang, Angelos Stavrou, and Anup Ghosh

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center for Secure Information Systems
George Mason University, VA, USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2013-0439

12. DISTRIBUTION / AVAILABILITY STATEMENT
For Public Release

13. SUPPLEMENTARY NOTES

{jwanga, astavrou, aghosh1}@gmu.edu

14. ABSTRACT

Over the past few years, virtualization has been employed to environments ranging from densely populated cloud computing clusters to home desktop computers. Security researchers embraced virtual machine monitors (VMMs) as a new mechanism to guarantee deep isolation of untrusted software components. Unfortunately, their widespread adoption promoted VMMs as a prime target for attackers. In this paper, we present HyperCheck, a hardware-assisted tampering detection framework designed to protect the integrity of VMMs and, for some classes of attacks, the underlying operating system (OS). HyperCheck leverages the CPU System Management Mode (SMM), present in x86 systems, to securely generate and transmit the full state of the protected machine to an external server. Using HyperCheck, we were able to ferret-out rootkits that targeted the integrity of both the Xen hypervisor and traditional OSes. Moreover, HyperCheck is robust against attacks that aim to disable or block its operation. Our experimental results show that HyperCheck can produce and communicate a scan of the state of the protected software in less than 40ms.

15. SUBJECT TERMS
Hypervisor, Protection framework, System Management Mode

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			aghosh1@gmu.edu
					19b. TELEPHONE NUMBER *(include area code)*

FINAL TECHNICAL REPORT

Autonomic Recovery of Enterprise Wide Systems After Attack of Failure With Forward Corrections

FA9550-07-1-0527

Principal Investigator

Anup Ghosh

Center for Secure Information Systems
George Mason University
Fairfax, VA

(MURI-07) Autonomic Recovery of Enterprise-Wide Systems After Attack or Failure with Forward Correction

MURI 07 THIRD YEAR REPORT

AFOSR Grant No: FA9550-07-1-0527
PRINCIPAL INVESTIGATOR: Professor Anup Ghosh
LEAD INSTITUTION: George Mason University
AFOSR PROGRAM MANAGER: Dr. Robert Herklotz, AFOSR, (703) 696-6565, robert.herklotz@afosr.af.mil

MOST RECENT GOVERNMENT REVIEW:
6 August 2010 – at University of Virginia, One-day technical review attended by the review team of Robert Herklotz (AFOSR).

PROGRAM OBJECTIVE: The objective of this Self Regenerative Incorruptible Enterprise topic is to develop technologies that will enable information systems to learn, regenerate themselves in response to unforeseen errors and/or attacks, and automatically improve their ability to deliver critical services. If successful, self-regenerative systems will reconstitute the information systems back to its initial operating capability while decreasing their vulnerability to an ever-increasing number of attacks. In this project the team is developing autonomic recovery and reconstitution techniques, which together it calls self-regeneration, for enterprise systems – clients and servers – for both incidental failures and deliberate attack. Enterprise computing systems need to be not only highly available, but also highly resistant to attack. Enterprise computing systems will fail due to software flaws, attacks, or hardware device failures. As a result, it is not adequate to assume that these systems will be able to always provide critical network services without building mechanisms that account for and handle these failures in service.

MURI CONSORTIUM RESEARCH TEAM MEMBERS:

- **Anup Ghosh, PI ,George Mason University**
- **Sushil Jajodia, George Mason University**
- **Angelos Keromytis, Columbia University**
- **Salvatore Stolfo, Columbia University**
- **Jason Nieh, Columbia University**
- **Peng Liu, Pennsylvania State University**

SCIENTIFIC APPROACH:

In this project, the team is developing autonomic recovery and regeneration mechanisms that will enable commodity systems to detect attacks, corruptions, and failures, then self regenerate to a known good state, for both program and data, while increasing the reliability and security of the software to be more resistant and less vulnerable to attack. The techniques allow for imperfect software systems to be deployed, but by providing autonomic recovery and regeneration, these systems will remain highly available and adaptive such that they will recover and be more resistant after attack. In this multi-disciplinary and multi-university team, they are developing a loosely coupled, layered defense system approach to system self-regeneration in two parts: whole system regeneration and application-based regeneration. The whole system approach uses machine virtualization in a transaction-based model to roll operating systems back to the last known good state prior to corruption or attack. The key innovation here is using a transaction-based model for commodity operating systems to enable consistent recovery and forward recovery with correction. The team is also developing application-specific self-regeneration techniques that employ verified error virtualization to enable a service to continue running even in the presence of faults and attacks that would otherwise crash the server or provide unauthorized privileged access. In deployment, the application-specific techniques will provide fine grained and frequent micro-recovery actions that keep servers functioning through software errors and attacks, while dynamically patching vulnerable software functions to render running software more resilient after attack or failure.

For *system self-healing*, the research aims to provide an isolation and application interaction-tracing framework. In

this system, all application interactions are recorded as transactions. The aim is to provide a recovery system that supports online recovery upon detection of a corruption without prohibitive performance and storage requirements. By recording application activities as container-based transactions, one trades the capability for taint tracking and replay within a container for lower processing and storage cost. The team has implemented a prototype, termed Journaling Computing System (JCS) using OpenVZ, a lightweight process virtualization mechanism.

With respect to *recovery of database systems after attack*, the team applies state machine theory, as long as the state transitions of an information system can be properly logged, damage assessment can be done via state classification (and dependence analysis), and recovery can be done via state roll-back/roll-forward operations. This work includes transactional database intrusion recovery to guarantee atomicity and consistency and development of a self-healing database management system (DBMS) to self-recover from malicious data corruptions without suffering from any downtime. This approach includes dynamic quarantine, fine-grained dependency analysis, and multi-version based online repairs.

With respect to *application self-healing*, the approach is based on the concept of software elasticity. Briefly, this refers to the ability of software to recover from faults (accidental or purposeful) under the right conditions, by translating exceptions caused by unforeseen failures into (possibly unrelated) error conditions that are handled by the software. The approach exploits this property by enabling software self-healing against a broad range of failures and attacks that allow legacy code to become elastic. In order to achieve this, the team leveraged mechanisms from a multitude of Computer Science disciplines, including programming languages, operating systems, machine learning, and graph theory. The primary challenges in this scheme revolve around balancing the 3-way tradeoff between:

- effectiveness of the self-healing mechanism (range of faults/attacks caught and mitigated)
- performance impact on the hardened software system
- safety of the recovery/self-healing operation

To that end, the team developed a series of novel mechanisms and algorithms that demonstrate the notion of "error virtualization using rescue points". In the work to date, the team has experimentally evaluated the effectiveness of their technique against real vulnerabilities and bugs, as well as against a much larger set of synthetic vulnerabilities; they have also characterized the overhead performance of the system showing it to be very low. The team has demonstrated the feasibility of creating completely automated self-healing software by building a fully integrated architecture and system, named ASSURE. This work has appeared in some of the top security and systems conferences (ASPLOS, USENIX Technical, IEEE Security & Privacy, NDSS, ACM-CCS).

MAJOR ACCOMPLISHMENTS TO-DATE:

The following major accomplishments have been accomplished to date:

- GMU developed a Moving Target Resilient Web Services framework. GMU is taking a proactive approach to defense by constantly shifting the attack surface of the web service in order to introduce uncertainty to the adversary. In particular, we leverage virtualization technologies to create virtual servers (VSs), each configured with a different software mix, producing diversified attack services that are rotated with a short time constant. Assuming N different VSs, $M < N$ will serve online requests at a time while offline servers are reverted to pre-defined pristine state. By constantly changing the set of M online servers and introducing randomness in their selection, adversaries of a web service will have to face multiple, unpredictable attack surfaces. While one cannot completely rule out their exploits, successful attacks against the web service will be more difficult and temporary as they get reverted to their pristine state. As a result, Web services will be significantly more resilient to attacks.

  Several challenging issues must still be addressed to materialize this vision. First, managing the complexity of many different software stacks is not straightforward, but we are developing techniques to address the technology management issues. We are developing a RESTful implementation of Web services that facilitate the construction of non-persistent system without disrupting web services. We will also consider the merits of a large number of diversified attack surfaces versus small and highly hardened ones. It is our belief that today's sophisticated web applications already create large attack surfaces. Making them unpredictable while constantly shifting their attack surface will raise the resilience of web services against intrusion. To prove our points in future, we will address the need of new metrics to measure service resilience.

- GMU implemented a secure desktop environment using lightweight virtualization and the Journaling

Computing System with recovery of tainted files. The GMU team designed a Linux-based secure desktop, called SecTop, that instantiates a lightweight virtual environment for every application in a pre-configured clean state. Using the JCS, application activities are recorded as state-based transactions. Performance evaluation studies demonstrated that acceptable overhead in processing and storage costs for desktop systems, while providing protection and recovery after attack. We recently coupled this desktop with a recovery system that is able to ascertain which files are corrupted, and selectively restore them back to their pre-infection state without loss of untainted data

- GMU developed a novel Hypervisor Integrity Monitoring technique. In order to assure that virtualized systems are not compromised themselves, additional protection for the integrity of the operating system and the virtualization execution environments is needed. To that end, we developed *HyperCheck*, a hardware-assisted tampering detection framework designed to protect the integrity of VMMs and, for some classes of attacks, the underlying operating system (OS). HyperCheck leverages the CPU System Management Mode (SMM), present in x86 systems, to securely generate and transmit the full state of the protected machine to an external server. Using HyperCheck, we were able to ferret-out rootkits that targeted the integrity of both the Xen hypervisor and traditional OSes. Moreover, HyperCheck is ro- bust against attacks that aim to disable or block its operation. Our experimental results show that HyperCheck can produce and communicate a scan of the state of the protected software in less than 40ms.

- GMU and Columbia developed a collaborative exchange network for network anomalies. We have demonstrated the collaborative deployment of network Anomaly Detection (AD) sensors. Our system examines the ingress http traffic and correlates AD alerts from two administratively disjoint domains: Columbia University and George Mason University. We have shown that, by exchanging packet content alerts between the two sites, we can achieve zero-day attack detection to inform local server instrumented systems that an attack has occurred with high confidence.

- GMU developed a scalable distributed data structure with recoverable encryption for cloud-based services. LH*RE is a new Scalable Distributed Data Structure (SDDS) for hash files stored in a cloud. The client-side symmetric encryption protects the data against the server-side disclosure. The encryption key(s) at the client are backed up in the file. The client may recover/ revoke any keys lost or stolen from its node. A trusted official can also do it on behalf of the client or of an authority, e.g., to imperatively access the data of a client missing or disabled. In contrast, with high assurance, e.g., 99%, the attacker of the cloud should not usually disclose any data, even if the intrusion succeeds over dozens or possibly thousands of servers for a larger file. Storage and primary key-based access performance of LH*RE should be about those of the well-known LH* SDDS. Two messages should typically suffice for a key-based search and four in the worst case, with the application data load factor of 70%, regardless of the file scale up. These features are among most efficient for a hash SDDS. LH*RE should be attractive with respect to the competition.

- Columbia University developed a lightweight virtual layered file systems for isolating untrusted applications from each other called Apiary. Desktop computers run many different applications, the compromise of any one of which can compromise the entire desktop given the lack of isolation among applications. Recovering a compromised desktop remains a time consuming task, which typically requires wiping everything and reinstalling the system from scratch. These security issues pose fundamental challenges as desktop computers are relied on for everything from financial transactions to medical records. To address these problems, we have created novel virtual layered file system (VLFS) technologies to improve system security. Unlike a traditional file system which is a monolithic entity, a VLFS dynamically composes together a set of software layers into a single file system view for a desktop. Changes to one layer are isolated and decoupled from changes to another. The VLFS dynamic composition feature enables powerful and easy-to-use security functionality. We are using VLFSes to build an architecture to enable security patches to be deployed effectively when managing large numbers of heterogeneously configured machines, and to speed system recovery from security exploits. We have also used VLFSes to develop a transparent desktop application fault containment architecture that is effective at limiting the damage from exploits to enable quick recovery while being as easy to use as a traditional desktop system.

- PSU developed PEDA: Cross-Layer Comprehensive Damage Assessment for Production Workload Server Systems. Damage assessment (or intrusion harm analysis) is a critical step in intrusion recovery of enterprise systems. However, for production workload server systems, the state of the art technologies can only do coarse-grained damage assessment at the system call level. (Coarse-grained damage assessment is

quite limited. In many cases, system call level damage assessment cannot identify all the damage (e.g., data corruption) that has been caused by an intrusion.) Doing fine-grained damage assessment at the instruction level is theoretically possible, but no existing fine-grained damage assessment technique is really practical for production workload servers: they usually cause 15-100x binary instrumentation overhead.

PEDA (Production Environment Damage Assessment), to the best of our knowledge, is the first practical fine-grained damage assessment technology for production workload server systems. Analyzing the harm of intrusion to enterprise servers is an onerous and error-prone work. Though dynamic taint tracking enables automatic fine-grained intrusion harm analysis for enterprise servers, the significant runtime overhead introduced is intolerable in the production workload environment. Our proposed system PEDA decouples the onerous analysis work from the online execution of the production servers and analyzes the "has-been-infected" execution during high fidelity replay on a separate instrumentation platform. Through a novel heterogeneous virtual machine migration technique, PEDA allows the online execution of servers to run atop fast hardware-assisted virtual machines (Xen) and the infected execution to be replayed atop binary instrumentation virtual machines (Qemu) for intrusion harm analysis. This approach can dramatically reduce the runtime overhead of online execution. In order to provide fine-grained taint seed to decoupled harm/damage analysis, PEDA bridges the gap between backward system call dependency tracking and forward intrusion taint analysis by integrating a one-step-forward auditing approach. Evaluation demonstrates the efficiency of PEDA system with runtime overhead as low as 5%, and the real-life intrusion studies successfully show the comprehensiveness and the precision of PEDA damage assessment.

PUBLICATIONS AND PRESENTATIONS:
12 papers in refereed journals,
41 papers in bound proceedings, and
31+ presentations at national and international meetings.

System & Database Self-Healing Publications:

1. "Experimental Results of Cross-Site Exchange of Web Content Anomaly Detector Alerts," Nathaniel Boggs, Sharath Hiremagalore, Angelos Stavrou, and Salvatore J. Stolfo. To appear in the *Proceedings of IEEE Conference on Homeland Security Technologies* (IEEE HST 2010), November 8-10, 2010, Waltham, MA, USA.

2. "HyperCheck: A Hardware-Assisted Integrity Monitor," Jiang Wang, Angelos Stavrou, and Anup K. Ghosh. In *13th International Symposium on Recent Advances in Intrusion Detection (RAID 2010)*, Ottawa, Canada, September 15-17, 2010.

3. "A Virtualization Architecture for In-Depth Kernel Isolation,"Jiang Wang, Sameer Niphadkar, Angelos Stavrou, Anup K. Ghosh. In Proceedings of 43rd Hawaii International International Conference on Systems Science, IEEE Computer Society, 5-8 January 2010, Koloa, Kauai, HI, USA.

4. "LH*RE: A Scalable distributed data structure with recoverable encryption," Sushil Jajodia, Witold Litwin, Thomas Schwartz, *Proc. 3rd IEEE International Conference on Cloud Computing* (Application and Industry Track), Miami, Florida, July 5-10, 2010.

5. "Encryption-based policy enforcement for cloud storage," Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, G. Pelosi, Pierangela Samarati, *Proc. 1st ICDCS Workshop on Security and Privacy in Cloud Computing* (ICDCS-SPCC),Genoa, Italy, June 21-25, 2010.

6. "On the Infeasibility of Modeling Polymorphic Shellcode: Re-thinking the Role of Learning in Intrusion Detection Systems," Y. Song, M E. Locasto, A. Stavrou , A.D. Keromytis and S. Stolfo, Machine Learning Journal, Special Issue on Adversarial Learning, 2009.

7. "Efficiently Tracking Application Interactions using Lightweight Virtualization," Yih Huang, Angelos Stavrou, Anup K. Ghosh and Sushil Jajodia. In the proceedings of the 1st Workshop on Virtualization Security (VMSec), in conjunction with ACM CCS 2008, October 2008.

8. "The Heisenberg Measuring Uncertainty in Lightweight Virtualization Testbeds," Quan Jia, Zhaohui Wang and Angelos Stavrou. In the *Proceedings 2nd Workshop on Cyber Security Experimentation and Test* (CSET '09). August, 2009, Montreal, Canada.

9. "Efficient Malware Containment Using Lightweight Virtualization," Jiang Wang, Angelos Stavrou, Anup Ghosh. In *Proceedings of 43rd Hawaii International Conference on System Sciences* (HICSS). IEEE Computer Society.

10. "Adaptive Anomaly Detection via Self-Calibration and Dynamic Updating", G. Cretu-Ciocarlie, A. Stavrou, M. Locasto and S. Stolfo, Proc. Int. Conf. on Recent Advanced in Intrusion Detection, RAID, 2009.

11. "Keep Your Friends Close: The Necessity for Updating an Anomaly Sensor with Legitimate Environment Changes", G. Cretu-Ciocarlie, A. Stavrou, M. Locasto and S. Stolfo, ACM Computer and Communications Security Conf. Workshop AI in Security, AISEC, 2009.

12. "Apiary: Easy-to-use Desktop Application Fault Containment on Commodity Operating Systems", Shaya Potter and Jason Nieh, *Proceedings of the 2010 USENIX Annual Technical Conference* (USENIX 2010), Boston, MA, June 23-25, 2010.

13. "Transparent, Lightweight Application Execution Replay on Commodity Multiprocessor Operating Systems", Oren Laadan, Nicolas Viennot, and Jason Nieh, *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS 2010), New York, NY, June 14-18, 2010.

14. "Shadow Honeypots," Michalis Polychronakis, Periklis Akritidis, Stelios Sidiroglou, Kostas G. Anagnostakis, Angelos D. Keromytis, and Evangelos Markatos. *In International Journal of Computer and Network Security* (IJCNS), vol. 2, no. 9, September 2010.

15. Peng Liu, Meng Yu, "Damage assessment and repair in attack resilient distributed database systems", Elsevier Computer Standards and Interfaces Journal, (2010), Accepted, in press.

16. Meng Yu, Wanyu Zang, Peng Liu, "Recovery of Data Integrity under Multi-Tier Architectures", IET Information Security, (2010), Accepted, in press.

17. S. Zhang, X. Jia, P. Liu, Cross-Layer Comprehensive Intrusion Harm Analysis for Production Workload Server Systems, Proc. 2010 Annual Computer Security Applications Conference (ACSAC), 2010, accepted.

18. X. Wang, Y. C. Jhi, S. Zhu, P. Liu, "Detecting Software Theft via System Call Based Birthmarks", Proc. 2009 Annual Computer Security Applications Conference (ACSAC), 2009.

19. X. Wang, Y. C. Jhi, S. Zhu, P. Liu, "Behavior Based Software Theft Detection," Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS), 2009.

20. Meng Yu, Hai Wang, Wanyu Zang, Peng Liu, "Evaluating Survivability and Costs of Three Virtual Machine based Server Architectures" , Proceedings of International Conference on Security and Cryptography, 2010

21. Hai Wang, Yan Su, Peng Liu, "A Semi-Markov Survivability Evaluation Model for Intrusion Tolerant Database Systems" , *Proceedings of ARES 2010* (The Fifth International Conference on Availability, Reliability and Security), 2010.

22. Shengzhi Zhang, Xi Xiong, Peng Liu, "Challenges in Improving the Survivability of Data Centers", *Proceedings of the Survivability in Cyberspace Workshop,* 2010.

23. Zhang, S., Xiong, X., Jia, X. and Liu, P. (2009) "Availability-sensitive Intrusion Recovery", *Proceedings of Second ACM Workshop on Virtual Machine Security,* Chicago, IL, November 2009, 6 page position paper

24. X. Xiong, X. Jia, P. Liu, "SHELF: Preserving Business Continuity and Availability in an Intrusion Recovery

System", *Proc. 2009 Annual Computer Security Applications Conference* (ACSAC), 2009

25. D. Kong, Y. C. Jhi, T. Gong, S. Zhu, P. Liu, H. Xi, "SAS: Semantics Aware Signature Generation for Polymorphic Worm Detection", *Proceedings of 2010 International ICST Conference on Security and Privacy in Communication Networks* (SECURECOMM), 2010, accepted.

26. Xiaoqi Jia, Xi Xiong, Jiwu Jing, Peng Liu, "Using Purpose Capturing Signatures to Defeat Computer Virus Mutating", *Proceedings of the Sixth International Conference on Information Security Practice and Experience Conference* (ISPEC), 2010.

27. D. Tian, X. Xiong, C. Hu, P. Liu, "Integrating Offline Analysis and Online Protection to Defeat Buffer Overflow Attacks," Proc. ISC 2010, LNCS, short paper, accepted.